



Applications in Practical High-End Computing - Group Project

Assignment - "Workflow"

Requirements

Supervisor: Dr Stuart Barnes

Authors: Mateusz Gołąb
Csaba Kerti
Jakub Kielbasa
Zsolt Kollarits

Course: CSTE / SETC

Table of contents

1.	Summary	3
a)	Communication with Customer	3
b)	Impact on following stages of project	3
2.	System's Dictionary	4
a)	Actors.....	4
b)	Layers	5
c)	Access layer	5
d)	Control layer	5
e)	Calculation layer	5
3.	User Requirements	6
a)	Functional requirements	6
b)	Non-functional requirements	6
4.	System Requirements	7
a)	System diagram	7
b)	Manage Simulation diagram.....	8
c)	Manage Modules diagram	11
d)	Manage Users diagram	13
e)	Recovery Mechanism diagram	15
5.	Final considerations and encountered problems	18
6.	Bibliography.....	18

1. Summary

Preparation of complete requirements list is a first and, according to many authorities, most difficult part during project life. Any misunderstanding or even inexactness can cause very serious complications in next parts of the project. Each pound saved on requirements analysis may come back as a dangerous difficulty in future and will cost tens times more.

Different methodologies choose different approaches to this issue. Some, older ones believe in inflexible boundaries of project parts, where after finishing requirements module comes design and amendments to previous part are not allowed. Others, more modern ones assume that changes in requirements may happen. In this group we can distinguish extreme ones which actually assume that changes in project specification are certain, and team members have to accept them and be able to give quick responses. Our team decided to use one of these (Agile) methodologies - Scrum.

a) Communication with Customer

Scrum assumes every-day meetings and very intensive communication with customer applying involving them into project ins and outs. This approach should decrease number of misunderstandings between project team and customer. Unfortunately, because of nature of academic assignments it was impossible to involve customer (lecturer) into every-day Scrums and long-lasting (usually 2-4 hours) discussions after them. Knowing this, we mocked Customer's presence on meetings by writing and receiving e-mails to and from them. This form of dialog, especially at the beginning helped us in defining specific requirements placed in next chapter in different form (functional, non-functional system).

b) Impact on following stages of project

Well-defined requirements are base of creating scalable and reasonable design. Knowing that we have tried to decrease amount of time spent on work in parallel on both of them. The problems were twosome, firstly Scrum itself assumes that requirements will change and secondly very short time period devoted for this assignment (less than 3 weeks) not allowed us to completely exhaust a subject. Second problem was independent from us so we decided to do our best within given time resources. For dealing with first issue we tried to distinguish project skeleton very early and focus on it, so that it would be fixed when design part starts and accept the fact that smaller features may come and go.

Requirements have also direct and indirect impact on test plan. Well-define use cases give possibility of creating reliable acceptance tests and traceability matrix.

Requirements' impact on implementation is only indirect (through design).

2. System's Dictionary

In order to avoid as many misunderstandings as possible, authors have decided to create System Dictionary. All, listed below, terms are being used in all following project's stages. Because of that fact, each change in it would cause many, difficult-to-detect complications, so we decided to put, especially lot of effort in creation of as short as possible but unambiguous definitions. For distinguishing them we adopted the notation with terms starting with capital letter.

a) Actors

User - person who uses System.
Scientist - person who can use Workflow to run simulation through Terminal using Client.
Administrator - person responsible for configuration of the Workflow.
Workflow Manager - program which controls Workflow. Responsible for sending commands and parameters to particular parts.
Recovery Manager - program which performs all tasks in order to assure safe recovery in case of crush during execution modules of Workflow Sequence.
DB Manager - program which controls DB's communication with Workflow.

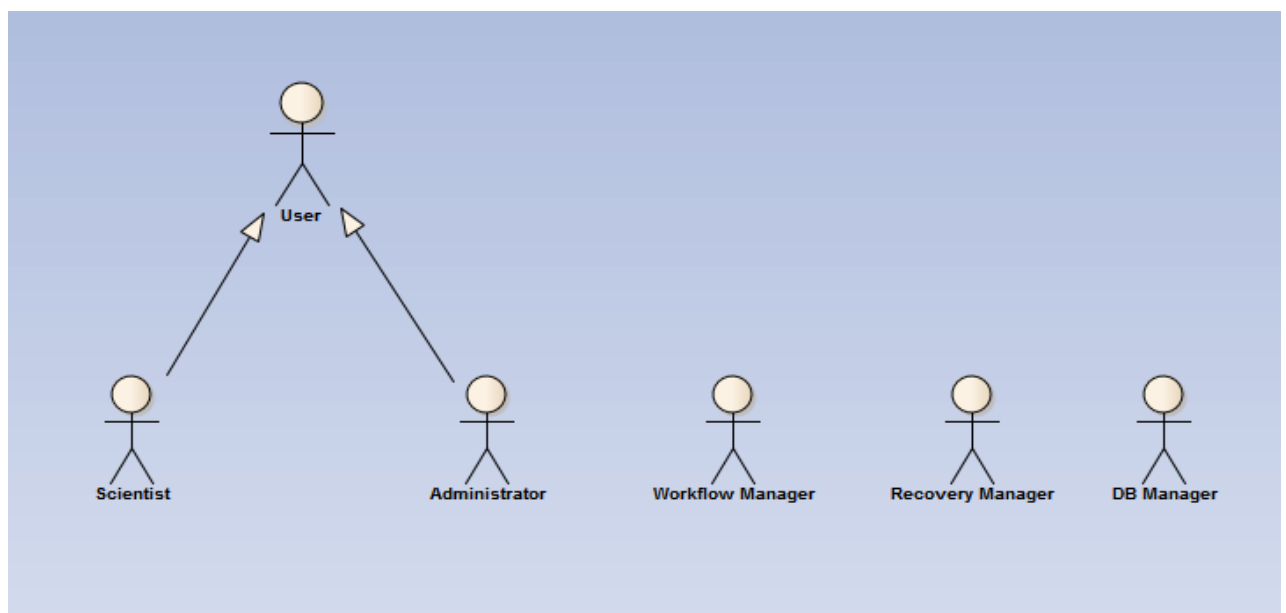


Diagram 1: dependencies between actors.

b) Layers

System - all resources and programs used in order to achieve project's aims. Organised, logically in three layers: access, controlling and calculating one.
--

c) Access layer

Client - program which gives Scientist possibility to remotely run simulation or Administrator to change settings of Workflow.

Terminal - place where Client runs.
--

d) Control layer

Workflow - all entities: programs and modules controlling and performing simulations scientist task.

Workflow Server - logical machine where Workflow runs.

Workflow Sequence - ordered, connected list of programs performing actual work of system. Sequence of steps necessary to finish task. Set up by Administrator.

Simulation - all start parameters needed in Workflow Sequence plus scientist id. May be considered as a instance of Workflow Sequence.

Simulations Queue - queue of Simulations in Workflow sent by Scientist waiting for execution.
--

Module - single unit-program within Workflow Sequence added by Administrator.
--

e) Calculation layer

Calculation Server - logical place where modules' calculations are being performed.
--

3. User Requirements

User requirements are functional and non-functional requirements of the system. They are dedicated for non-technical persons, usually customers, therefore there have to be prepared in natural language. Main point of them is to keep requirements as a short list - over specification is one of most common reason of project failures.

a) Functional requirements

Usually they are explicit users' requirements. They simply describe wanted features of the system.

FU1: User can add/remove arbitrary number of modules into workflow.
FU2: User can run simulation with uploaded parameters.
FU3: Recovery system: possibility of restarting workflow (from the last stable/good point) when system crushes.
FU3.1: Monitoring of errors: Users can see the exact location of failures.
FU3.2: Flexible recovery policy (depending on expected time of execution we decide to store data before/after module or after each iteration)
FU4: Many users have possibility to connect to system simultaneously. But there is only one running program at time (users requests' go to queue - serial workflow).

b) Non-functional requirements

They are very often implicit, hidden from user but absolutely necessary for system performance.

NF1: Reliability/Validation: take care of input/output formats of specific modules.
NF2 : Use universal/independent communication standards between modules
NF3: Data flow should be based on XML files.
NF4: Modules can run on many different remote systems using different platforms but they have to be already installed.
NF5: Network connection between terminal, workload server and remote systems are established.

4. System Requirements

a) System diagram

System diagram illustrates logical structure of System. It is built based on few smaller sub-systems and several actors. Part of actors are external, physical persons like Scientist or Administrator but there are also internal ones who represents System's programs. As we can see on diagram below most of use cases have been gathered in groups - only two, very simple ones are specified on the level of main System.

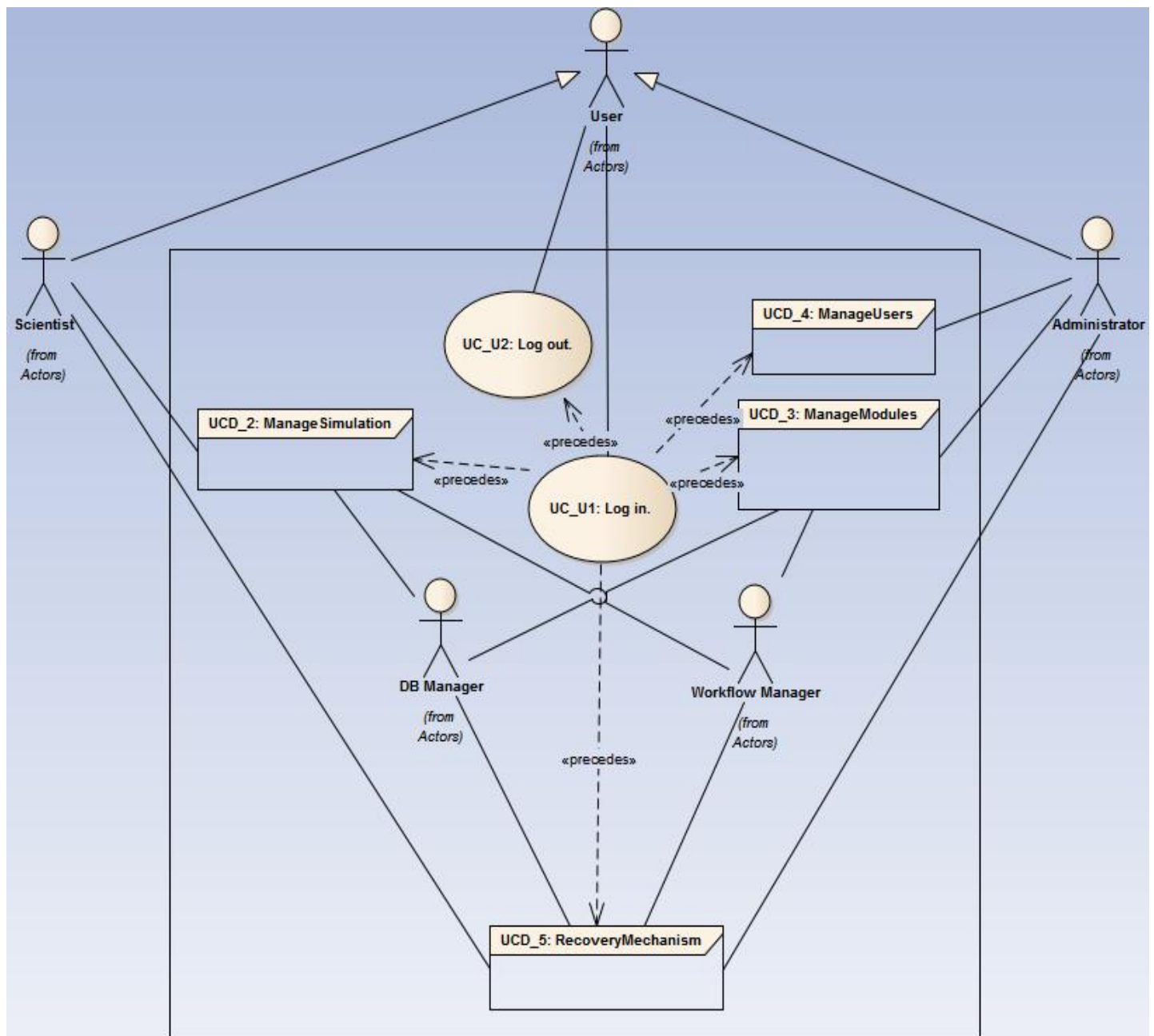


Diagram 2: use case diagram of entire System.

- **UC_U1: Log in.**

Description:	User can log in to System using Client.
Requirements:	1. User should gain access to System.
Constraints:	-
Basic Path:	<i>Successful log in.</i> 1. User connects through client to system. 2. User types credentials on log in screen. 3. User gain access to system resources.
Alternate Path:	<i>Wrong Credentials</i> 3a. User sees message with information that credentials are wrong. Join with 2 Basic Path.

- **UC_U2: Log out.**

Description:	User can log out from System using Client.
Requirements:	1. User should safely leave System.
Constraints:	1. Successful log in. (Pre-condition)
Basic Path:	<i>Successful log out.</i> 1. User logs out from system.

b) Manage Simulation diagram

Sub-diagram, represents part of System responsible for starting, stopping and operating on simulation. From outside, ideally, it should be accessible only for Scientist. Internally, there will be Managers, who have access to its resources and fulfil Scientist requirements.

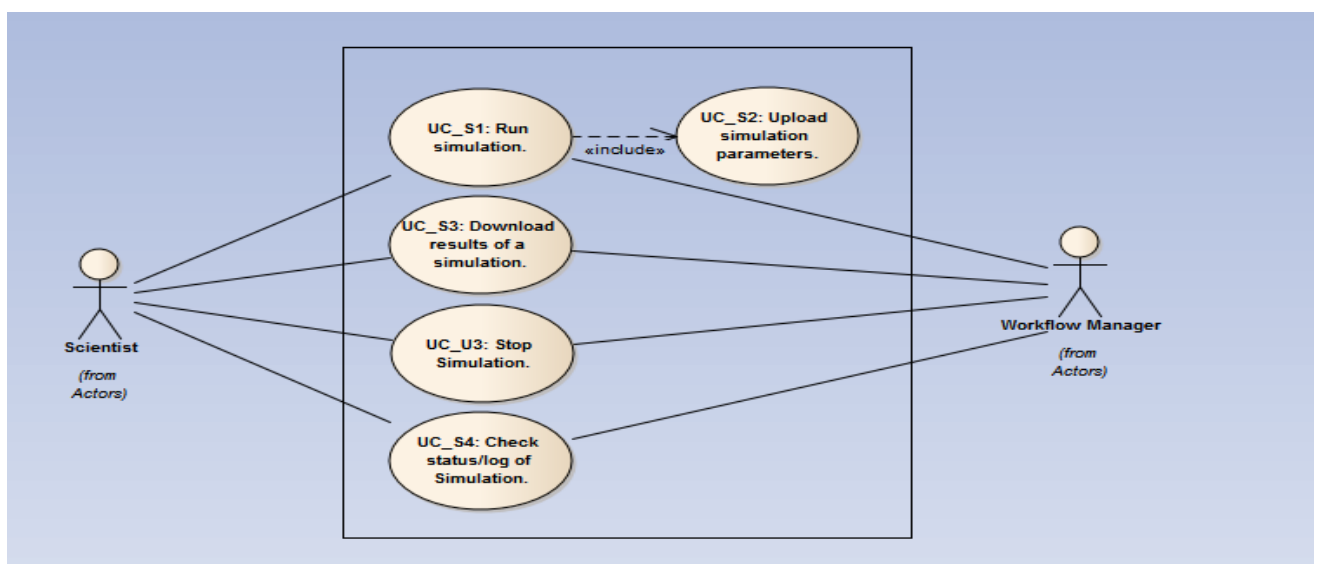


Diagram 3: use case diagram of Simulations' Management.

- **UC_S1: Run Simulation.**

Description:	Scientist connects through terminal and runs customized simulation.
Requirements:	1. User get information that their simulation has been added to queue or started.
Constraints:	1. Successful log in. (Pre-condition)
Basic Path:	<i>Immediate start.</i> 1. Scientist choose kind of workflow them want to run. 2. Scientist is redirected to upload parameter screen. (Jump to UC_S2). 3. After successful upload Scientist run simulation. 4. Workflow Manager retrieves information and sends back information whether immediate start is possible. 5. Scientist get information that simulation run has started.
Alternate Path:	<i>Added to queue</i> 5a. User get information that their simulation has been added to queue.

- **UC_S2: Upload Simulation parameters.**

Description:	Scientist upload necessary parameters for simulation's start.
Requirements:	1. Upload correct parameters of simulation's start.
Constraints:	1. Successfully going through UC_S1. (Pre-condition)
Basic Path:	<i>Correct uploading</i> 1. Scientist upload execution parameters. 2. Scientist confirms upload 3. Scientist get message that uploading was successful
Alternate Path:	<i>Wrong parameters</i> 3a. User gets information that one/many parameters are inappropriate. 4a. User can input correct parameters again

- **UC_S3: Download results of a Simulation.**

Description:	Scientist downloads results of simulation's run.
Requirements:	1. Download results of simulation in correct format etc.
Constraints:	1. Successfully going through UC_S1. (Pre-condition)
Basic Path:	<i>Successful download</i> 1. Scientist connects to download page. 2. Scientist chooses wanted results. 3. Workflow Manager let Scientist downloads results.
Alternate Path:	<i>No results to download</i> 2a. There are no results to being downloaded. 3a. Redirected to main screen.

- **UC_U3: Stop Simulation.**

Description:	User wants to immediately stop simulation run using Client.
Requirements:	1. Successful stop simulation.
Constraints:	1. Successfully log in. (Pre-condition)
Basic Path:	<i>Selected operation is executable</i> 1. User decides to stop the Simulation. 2. Workflow Manager checks the availability of the selected operation. 3. If the stopping is available, the system executes it.
Alternate Path:	<i>Stopping already started by another User.</i> 2a. Workflow Manager informs User that Simulation has already been stopped. 3a. Redirected to main screen.

- **UC_S4: Check status/log of Simulation.**

Description:	Scientist wants to check status of their Simulation.
Requirements:	1. Gain information about Simulation state.
Constraints:	1. Successfully log in. (Pre-condition)
Basic Path:	<i>Successful status.</i> 1. The Scientist requires the status of workflow 2. Workflow Manager prints out the status of Simulation.

c) Manage Modules diagram

Sub-diagram, responsible for creating Workflow Sequence. Externally it is only accessible to Administrator, internally there are Managers which executes required operations in Workflow in order to achieve Administrator's aim.

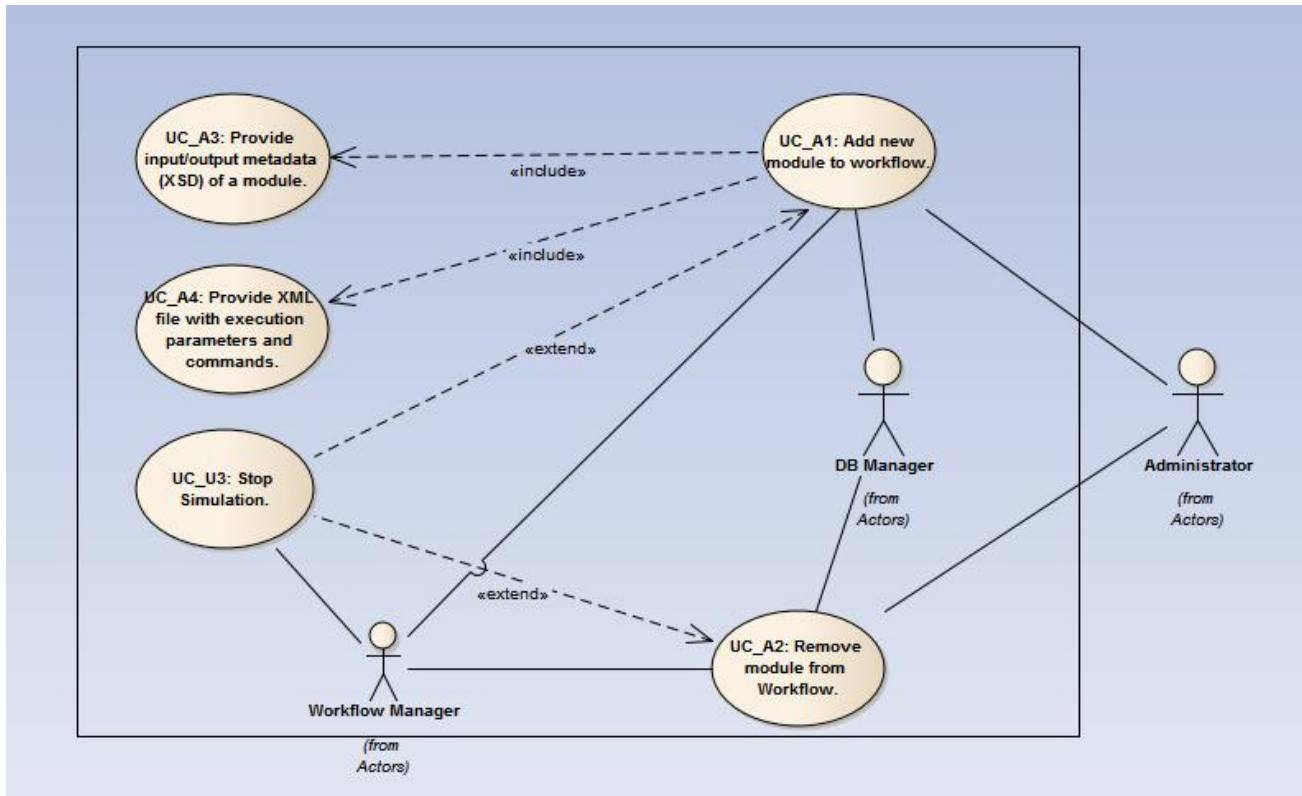


Diagram 4: use case diagram of Modules' Management.

- **UC_A1: Add new module to Workflow.**

Description:	Administrator wants to add module to existing Workflow Sequence
Requirements:	1. Successful addition of new module.
Constraints:	1. Successfully log in. (Pre-condition)
Basic Path:	<p><i>Successful addition.</i></p> <ol style="list-style-type: none"> 1. The Administrator chooses to add new module to Workflow Sequence. 2. If a Simulation is running then Workflow Manager invokes UC_U3. 3. A window appear, where the Administrator can set the properties of the module

	4. UC_A3 is invoked by Workflow Manager. 5. UC_A4 is invoked by Workflow Manager. 6. DB Manager stores data about Module in DB. 7. Administrator retrieves information about successful addition.
--	--

- **UC_A2: Remove module from Workflow.**

Description:	Administrator wants to remove Module from existing Workflow Sequence.
Requirements:	1. Secure remove of Module.
Constraints:	1. Successfully log in. 2. Exists at least 1 Module.
Basic Path:	<i>Successful removal</i> 1. A confirmation dialog box appears to get permission to remove the appropriate Module. 2. The Administrator chooses the confirming button. 3. If a Simulation is running then Workflow Manager invokes UC_U3. 4. DB Manager remove Module's information from DB. 5. Administrator retrieves information about successful removal.

- **UC_A3: Provide input/output metadata (XSD) of a Module.**

Description:	Administrator needs to provide input/output XSD files as a part of adding new Module.
Requirements:	1. Secure uploading XSD files to DB.
Constraints:	1. Successfully going through UC_A1. (Pre-condition)
Basic Path:	<i>Successful upload</i> 1. The select the appropriate XSD descriptor file from the file system using a file browser window for provide the input validator. 2. Workflow Manager checks the validity of the input XSD file 3. The select the appropriate XSD descriptor file from the file system using a file browser window for provide the input validator. 4. The Workflow Manager checks the validity of the output XSD file 5. Administrator retrieves information about successful uploading.

- **UC_A4: Provide XML file with execution parameters and commands.**

Description:	Administrator needs to provide XML files with starting parameters as a part of adding new Module.
Requirements:	1. Secure uploading XML files with parameters to DB.
Constraints:	1. Successfully going through UC_A1. (Pre-condition)
Basic Path:	<p><i>Successful upload</i></p> <ol style="list-style-type: none"> 1. The select the appropriate XML descriptor file from the file system using a file browser window for provide the descriptor file. 2. Workflow Manager checks the validity of the input XML file. 3. Administrator confirms decision. 4. DB Managers stores commands in DB. 5. Administrator retrieves information about successful uploading.

d) Manage Users diagram

Sub-diagram, responsible for managements of users' accounts. Actor, externally responsible for this activities is Administrator. There is also, internal Manager which stands for storing data about Users into data base.

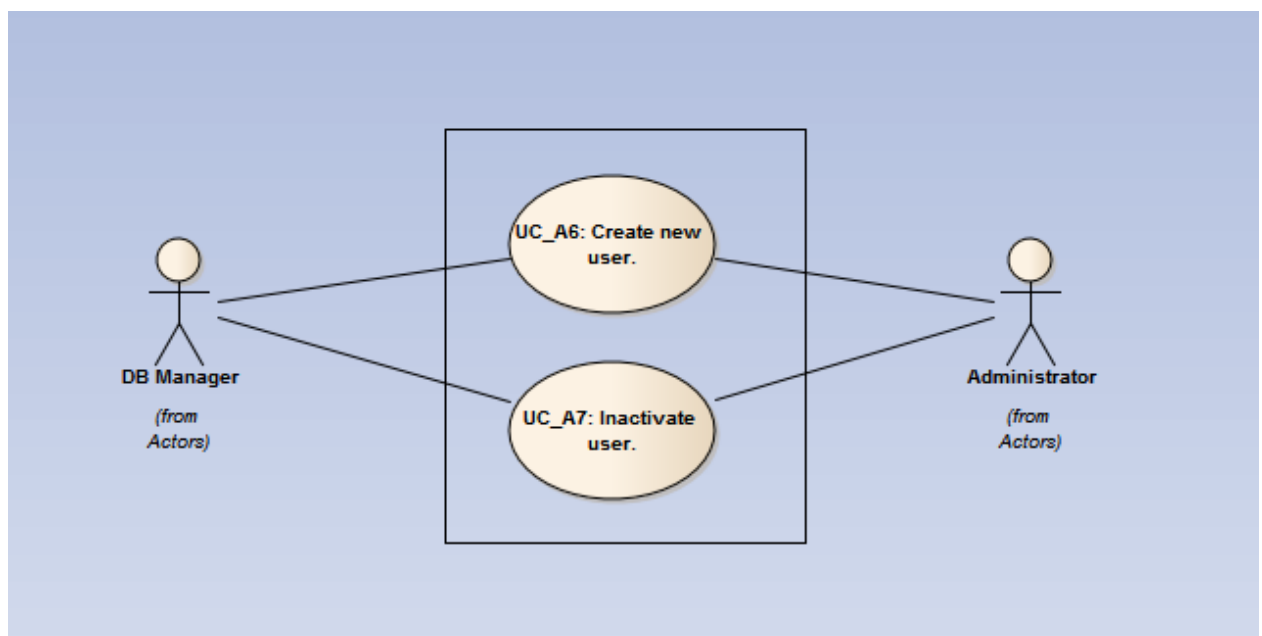


Diagram 5: use case diagram of Users' Management.

- **UC_A6: Create new User.**

Description:	Administrator creates new User account using Client.
Requirements:	1. New User's account created.
Constraints:	1. Successfully log in. (Pre-condition)
Basic Path:	<i>Success addition</i> 1. Administrator chooses option "Add user" 2. Administrator puts correct user details 3. Administrator confirms adding a new user. 4. DB Manager stores new user in DB. 5. Workflow provides information to the Administrator about successfully added user.
Alternate Path:	<i>New user data is incorrect.</i> 4a. Administrator retrieves information about wrong User data. 5a. Redirected to main screen.

- **UC_A7: Inactivate User.**

Description:	Administrator inactivates existing User account using Client.
Requirements:	1. User's account inactivated completely.
Constraints:	1. Successfully log in. 2. Exists at least 1 User.
Basic Path:	<i>Success inactivation</i> 1. Administrator chooses "inactivate user option" 2. Administrator chooses user to inactivate. 3. Administrator confirms user inactivation. 4. DB Manager inactivates user (change status in DB). 5. Workflow provides information to the Administrator about successfully inactivation of a User.
Alternate Path:	<i>User's simulation is running</i> 4a. Administrator retrieves information that it is impossible to inactivate User as long as their Simulation is running. 5a. Redirected to main screen.

e) Recovery Mechanism diagram

Sub-diagram, responsible for, probably most important part of System - Recovery mechanism. All internal Managers are used here for assuring full access to all resources needed. The only actor from outside is Administrator who is responsible for setting mechanism. Ideally all recovery should be fully automatic after this.

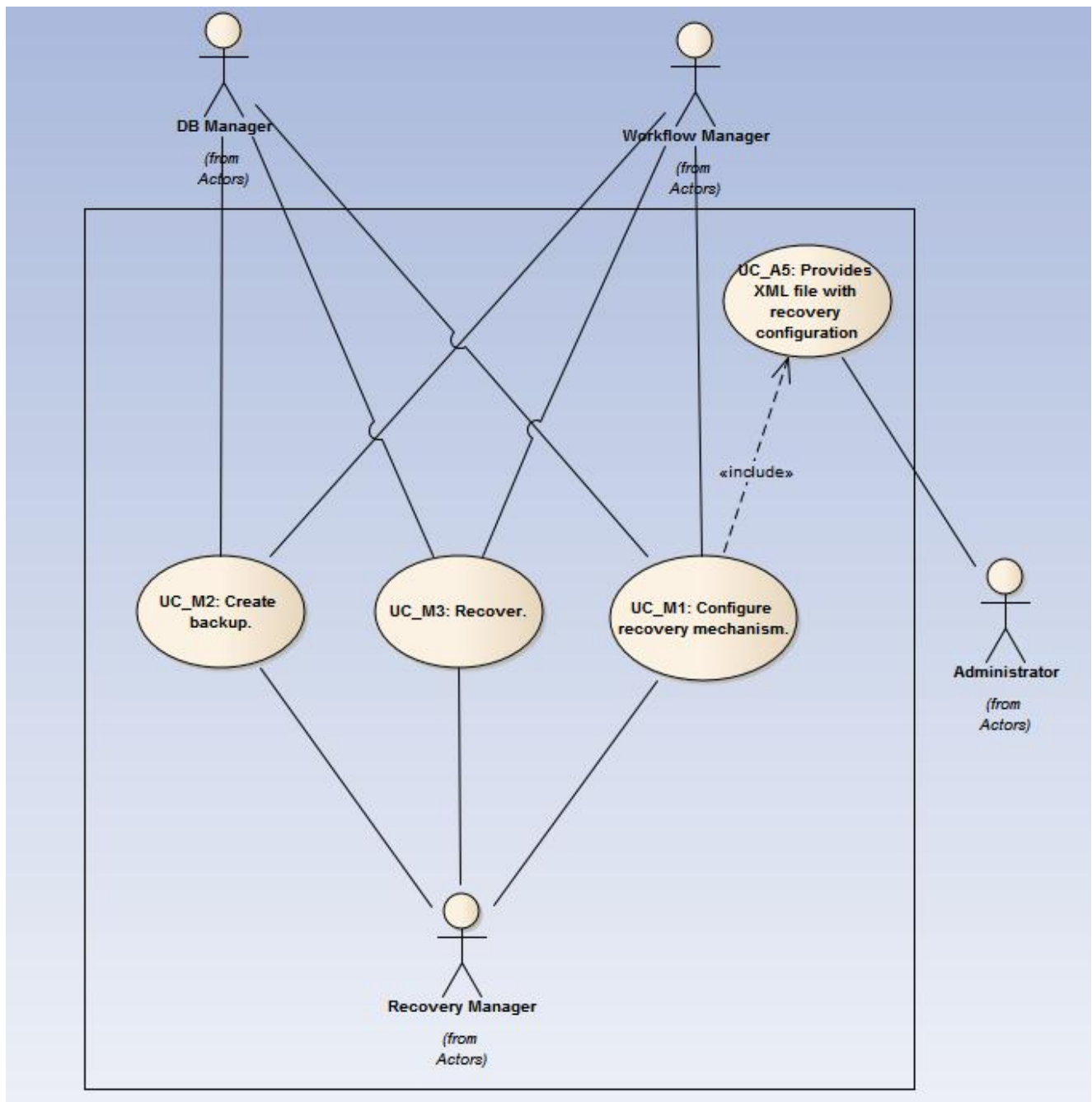


Diagram 6: use case diagram of Recovery Mechanism.

- **UC_A5: Provides XML file with recovery configuration**

Description:	Administrator provides all needed variables for configuration of recovery process.
Requirements:	1. All needed settings provided.
Constraints:	1. Successful log in. (Pre-condition) 2. Successfully going through UC_M1. (Pre-condition).
Basic Path:	<i>Successful providing</i> 1. Administrator chooses option "upload configuration file" 2. Administrator chooses .xml file from his disk. 3. Administrator confirms uploading new configuration file. 4. Workflow Manager provides Information about successfully changed configuration.

- **UC_M1: Configure recovery mechanism.**

Description:	Recovery Manager wants to configure recovery environment after creation of Workflow Sequence.
Requirements:	1. Fully prepared recovery mechanism prepared.
Constraints:	1. Workflow Sequence exists. (Pre-condition)
Basic Path:	<i>Successful configuration</i> 1. Workflow Manager starts preparing structure of recovery configuration. 2. Waiting for UC_A5 to finish. 3. Recovery Manager uses Administrator parameters for automatic set up of variables. 4. Workflow Manager sends data through DB Manager to data base.

- **UC_M2: Create backup.**

Description:	Recovery Manager wants to create backup after finishing each relevant Module.
Requirements:	1. Fully prepared backup stored in DB.
Constraints:	1. UC_S1 is running.
Basic Path:	<i>Successful backup.</i>

	<ol style="list-style-type: none"> 1. Workflow Manager detects that Module finished its work. 2. If Module is supposed to be backup then Workflow Manager sends information to Recovery Manager about that. 3. Recovery Manager creates backup of current Simulation stage. 4. Recovery Manager connects with DB Manager which saves data into DB. 5. DB Manager informs Recovery Manager about successful transaction. 6. Recovery Manager informs Workflow Manager about successful backup. 7. Workflow Manager runs another Module.
Alternate Path:	<i>Do not create a Backup.</i> 2a. Module do not need to be backup, finish. <GO TO 7>
Exception:	<i>DB exception.</i> 4b. Try again after specified amount of time. <GO TO 3>

- **UC_M3: Recover.**

Description:	Recovery Manager wants to restore backup after modules crush.
Requirements:	1. Correctly recovered System.
Constraints:	<ol style="list-style-type: none"> 1. UC_S1 is running. 2. UC_M2 is working correctly.
Basic Path:	<i>Successful recovery.</i> <ol style="list-style-type: none"> 1. Workflow Manager notices that simulation crushes and sends information to Recovery Manager. 2. Recovery Manager sends information to DB Manager about which backup is needed. 3. DB Manager sends backup files to Recovery Manager. 4. Recovery Manager sends backup to Workflow Manager. 5. Workflow Manager runs again Simulation with given backup. 6. Simulation run correctly
Alternate Path:	<i>Recovery failure.</i> <ol style="list-style-type: none"> 6a. Workflow Manager sends information to Recovery Manager about failure. 7a. Recovery Manager depending on recovery parameters changes or not ID of needed backup to download. <GO TO 2>

Exception:	<p><i>Too many failures..</i></p> <p>6b. Workflow Manager sends information to Recovery Manager about failure.</p> <p>7b. Recovery Manager sends back information to Workflow Manager that it should stop Simulation because there is no point in further recovering.</p> <p>8b. Workflow Manager stops Simulation. <END></p>
-------------------	---

5. Final considerations and encountered problems

Requirements change in time. This sad, but true sentence is responsible for most of problems in this part of project. Situation become more complicated when time pressure is bigger and number of dependencies in project increase. Situation become even more complicated when employees have to work simultaneously on two very serious projects.

Problem mentioned above caused situation were authors had sometimes to change requirements after officially closure of this stage. It was very problematic situation but, thanks to Scrum methodology it was easier to deal with it.

Technically, the biggest problems were lacks in knowledge of UML. Not once authors spent a lot of time going through tutorial about this language. Final result for sure is not perfect, but we want to believe that it is acceptable.

6. Bibliography

[1] UML Distilled: A Brief Guide to the Standard Object Modelling Language

Fowler M.:Addison-Wesley Professional. 2000

[2] Software Engineering,

Sommerville I.: Person Education, Inc. 2011

[3] Sparx Tutorials

<http://www.sparxsystems.com/resources/index.html>

[4] Wikipedia

<http://en.wikipedia.org/>